Hirotaka YOSHIDA et al

日本国特許庁
JAPAN PATENT OFFICE

McDermott Will & Emery LLP

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。
This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２００３年　６月　３日

出願番号
Application Number: 特願２００３－１５７４４４

[ST. 10/C]: [ＪＰ２００３－１５７４４４]

出願人
Applicant(s): 株式会社日立製作所

CERTIFIED COPY OF
PRIORITY DOCUMENT

BEST AVAILABLE COPY

２００４年　２月２５日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫

出証番号　出証特２００４－３０１３３４６

【書類名】　特許願
【整理番号】　K03003151A
【あて先】　特許庁長官殿
【国際特許分類】　G09C　1/00
【発明者】
　【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所システム開発研究所内
　【氏名】　吉田　博隆
【発明者】
　【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所システム開発研究所内
　【氏名】　古屋　聡一
【特許出願人】
　【識別番号】　000005108
　【氏名又は名称】　株式会社　日立製作所
【代理人】
　【識別番号】　100075096
　【弁理士】
　【氏名又は名称】　作田　康夫
【手数料の表示】
　【予納台帳番号】　013088
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　改竄検知可能な、共通鍵暗号の暗号化装置または復号化装置

【特許請求の範囲】

【請求項１】

共通鍵暗号の暗号化装置であって，

冗長データとメッセージとからなる平文を特定の長さで区切った複数の平文ブロックを生成する処理部と，

秘密鍵から，前記平文よりも長い乱数列を生成し，前記乱数列から前記平文ブロックに対応する暗号化のための乱数ブロックを生成し，前記平文ブロックと前記乱数ブロックを用いて暗号文ブロックを暗号演算する処理部と，

前記乱数列から前記暗号文ブロックに対応する認証のための乱数ブロックを生成し，暗号文ブロックと前記乱数ブロックを用いてメッセージ認証子ブロックを認証演算する処理部とを備える共通鍵暗号の暗号化装置。

【請求項２】

請求項1記載の共通鍵暗号の暗号化装置であって，

前記暗号演算処理部と前記認証演算処理部は一つ以上の，その合計の長さは，前記平文ブロックの合計より長く平文ブロックの合計の2倍より短い前記乱数ブロックを用いる共通鍵暗号の暗号化装置。

【請求項３】

請求項２記載の共通鍵暗号の暗号化装置であって，

前記暗号演算処理部は，前記平文ブロックを用いた2項演算や単項演算を決められた手順に従って1回以上行い，

前記認証演算処理部は，前記暗号文ブロックを用いた2項演算や単項演算を決められた手順に従って1回以上行い，

得られた複数の暗号文ブロックとメッセージ認証子ブロックを組み合わせ，暗号文として出力する処理部を備える共通鍵暗号の暗号化装置。

【請求項４】

請求項２記載の共通鍵暗号の暗号化装置であって，

前記暗号演算処理部は、前記暗号演算を排他的論理和によって行い，

前記認証演算処理部は、前記認証演算を算術乗算と算術加算により行う共通鍵暗号の暗号化装置。

【請求項５】

請求項２記載の共通鍵暗号の暗号化装置であって，

前記暗号演算処理部は、前記暗号演算を排他的論理和によって行い，

前記認証演算処理部は、前記認証演算を有限体上の乗算と算術加算により行う共通鍵暗号の暗号化装置。

【請求項６】

請求項２記載の共通鍵暗号の暗号化装置であって，

前記暗号演算処理部と前記認証演算処理部とは、使う乱数ブロックを共有する共通鍵暗号の暗号化装置。

【請求項７】

請求項２記載の共通鍵暗号の暗号化装置であって，

前記暗号演算処理部と前記認証演算処理部とは、使う乱数ブロックを共有する共通鍵暗号の暗号化装置。

【請求項８】

請求項２記載の共通鍵暗号の暗号化装置であって，

前記暗号演算処理部と前記認証演算処理部とは、異なる乱数ブロックを使用する共通鍵暗号の暗号化装置。

【請求項９】

請求項２記載の共通鍵暗号の暗号化装置であって，

前記秘密鍵から前記乱数列を生成する擬似乱数生成処理部を備える共通鍵暗号の暗号化装置。

【請求項１０】

請求項９の共通鍵暗号の暗号化装置であって，

前記メッセージを複数に分割する処理部を備え，

前記擬似乱数生成処理部は，前記分割数分の乱数列を生成し，

前記分割したメッセージと前記乱数列のいずれかをそれぞれ異なる演算ユニットに割り当て，並列処理を行わせる処理部を備える共通鍵暗号の暗号化装置。

【請求項１１】

共通鍵暗号の復号化装置であって，

暗号文を特定の長さで区切った複数の暗号文ブロックを生成する処理部と，

秘密鍵から，前記暗号文よりも長い乱数列を生成し，前記乱数列から前記暗号文ブロックに対応する認証のための乱数ブロックを生成し，暗号文ブロックと前記乱数ブロックを用いてメッセージ認証子ブロックを認証演算する処理部と，

前記乱数列から暗号文ブロックに対応する復号のための乱数ブロックを生成し，前記暗号文ブロックと，前記乱数ブロックを用いて平文ブロックを復号演算する処理部とを備える共通鍵暗号の復号化装置。

【請求項１２】

請求項１１記載の共通鍵暗号の復号化装置であって，

前記認証演算処理部と前記復号演算処理部は一つ以上の，その合計の長さは，前記平文ブロックの合計より長く平文ブロックの合計の2倍より短い前記乱数ブロックを用いる共通鍵暗号の復号化装置。

【請求項１３】

請求項１２記載の共通鍵暗号の復号化装置であって，

前記平文ブロックを複数連結して平文を生成する処理部と，

前記平文に含まれる冗長データを抽出する処理部と，

前記冗長データを検査し，前記暗号文への改ざんの有無を検出する処理部とを備える共通鍵暗号の復号化装置。

【請求項１４】

コンピュータに，共通鍵暗号の暗号化処理を実行させるプログラムを記憶した媒体であって，

前記プログラムは，前記コンピュータに，

冗長データとメッセージとからなる平文を特定の長さで区切った複数の平文ブロックを生成させ，

秘密鍵から，前記平文よりも長い乱数列を生成し，前記乱数列から前記平文ブロックに対応する暗号化のための乱数ブロックを生成し，前記平文ブロックと前記乱数ブロックを用いて暗号文ブロックを暗号演算させ，

前記乱数列から前記暗号文ブロックに対応する認証のための乱数ブロックを生成させ，暗号文ブロックと前記乱数ブロックを用いてメッセージ認証子ブロックを認証演算させる、プログラムを記憶した媒体。

【請求項１５】

請求項１４のプログラムを記憶した媒体であって，

前記暗号演算と前記認証演算は一つ以上の，その合計の長さは，前記平文ブロックの合計より長く平文ブロックの合計の2倍より短い前記乱数ブロックを用いる、プログラムを記憶した媒体。

【請求項１６】

請求項１５のプログラムを記憶した媒体であって，

前記暗号演算として，前記平文ブロックを用いた2項演算や単項演算を決められた手順に従って1回以上行わせ，

前記認証演算として，前記暗号文ブロックを用いた2項演算や単項演算を決められた手順に従って1回以上行わせ，

得られた複数の暗号文ブロックとメッセージ認証子ブロックを組み合わせ，暗号文として出力させる、プログラムを記憶した媒体。

【請求項１７】

請求項１５のプログラムを記憶した媒体であって，

前記暗号演算を排他的論理和によって行わせ，

前記認証演算を算術乗算と算術加算により行わせる、プログラムを記憶した媒体。

【請求項１８】

請求項１５のプログラムを記憶した媒体であって，

前記暗号演算を排他的論理和によって行わせ，

前記認証演算を有限体上の乗算と算術加算により行わせる、プログラムを記憶した媒体。

【請求項１９】

請求項１５のプログラムを記憶した媒体であって，

前記暗号演算と前記認証演算とが使う乱数ブロックを共有させる、プログラム

を記憶した媒体。

【請求項２０】

請求項１５のプログラムを記憶した媒体であって，

前記暗号演算と前記認証演算とが使う乱数ブロックを共有させる、プログラムを記憶した媒体。

【請求項２１】

請求項１５のプログラムを記憶した媒体であって，

前記秘密鍵から前記乱数列を生成させる擬似乱数生成処理を行わせる，プログラムを記憶した媒体。

【請求項２２】

請求項２１のプログラムを記憶した媒体であって，

前記メッセージを複数に分割し，

前記擬似乱数生成処理によって，前記分割数分の乱数列を生成させ，

前記分割したメッセージと前記乱数列のいずれかをそれぞれ異なる演算ユニットに割り当て，並列処理を行わせる、プログラムを記憶した媒体。

【請求項２３】

コンピュータに，共通鍵暗号の復号化処理を実行させるプログラムを記憶した媒体であって，

前記プログラムは，前記コンピュータに，

暗号文を特定の長さで区切った複数の暗号文ブロックを生成させ，

秘密鍵から，前記暗号文よりも長い乱数列を生成させ，前記乱数列から前記暗号文ブロックに対応する認証のための乱数ブロックを生成させ，暗号文ブロックと前記乱数ブロックを用いてメッセージ認証子ブロックを認証演算させ，

前記乱数列から暗号文ブロックに対応する復号のための乱数ブロックを生成させ，前記暗号文ブロックと，前記乱数ブロックを用いて平文ブロックを復号演算させる、プログラムを記憶した媒体。

【請求項２４】

請求項２３のプログラムを記憶した媒体であって，

前記復号演算と前記認証演算とに一つ以上の，その合計の長さは，前記平文ブ

ロックの合計より長く平文ブロックの合計の2倍より短い前記乱数ブロックを用いらせる、プログラムを記憶した媒体。

【請求項２５】

請求項２４のプログラムを記憶した媒体であって，

前記平文ブロックを複数連結して平文を生成させ，

前記平文に含まれる冗長データを抽出させ，

前記冗長データを検査させ，前記暗号文への改ざんの有無を検出させる、プログラムを記憶した媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は，秘密情報のセキュリティを確保する技術に関する。

【０００２】

【従来の技術】

従来の暗号処理装置は，データを秘匿する目的のブロック暗号やストリーム暗号が提案されていた。ブロック暗号にはAES（Advanced Encryption Standard）を始め，いろいろなアルゴリズムが提案されている。

【０００３】

ブロック暗号はECB，CBC，CFB，OFB，カウンタモードなどのブロック暗号操作モードにより全体の暗号処理の安全性や性質を議論するが，これまでに暗号化処理と改ざん検出を同時に行う操作モードは，iaPCBCモードが知られているだけで，残りのモードは改ざんの検出がそれ自身では不可能である。iaPCBCモードは非特許文献１で扱っている。

【０００４】

iaPCBCモードはブロック暗号を用いた操作モードであり，暗号化処理では並列処理，事前計算などができないので，高速な処理が要求される環境への実装が困難であった。

【０００５】

これに対して，メッセージ認証子と呼ばれる改ざん検出のための暗号学的チェ

ックサム(以下，MACという)を生成する方法が提案され，ブロック暗号の上記操作モードの暗号処理でも必要に応じてMAC生成処理を同時にかつまったく独立の機構として実装することで，暗号処理と改ざん検出が同時に可能となった。しかし，この場合にはまったく独立な暗号学的鍵を2度，すなわち，暗号化用と改ざん検出用，共有する必要があるという点，それから暗号化されるデータを2度処理，つまり，暗号化処理とMAC生成処理，にかける必要があり，システムが複雑になったり，長いデータの処理に向かなくなったりなどの懸念があった。さらにブロック暗号の処理速度が現在の通信の速度に比べて低速であり，これらブロック暗号とMACの組み合わせ技術は，ギガビットやテラビット処理といった高速処理が要求される用途への応用が困難であった。

【０００６】

MACと軽い処理との組み合わせが操作モードを実現できることが知られていた。それをモードとして用いたストリーム暗号は，暗号処理と改ざん検出を同時に行えるほか，上記ブロック暗号の処理にくらべて，2倍～20倍の比率でストリーム暗号がより高速であるが，どのMAC生成方法についても，ブロック暗号とMACの組み合わせ同様，メッセージの長さに対してその2倍の長さの擬似乱数を必要とし，必要な乱数生成に時間がかかったり，ひとつのメッセージについて2度の処理を行う必要などがあった。

【０００７】

より詳しくMAC生成方法を考えると，本来のストリーム暗号に，付帯的に必要となるメカニズムや計算量が非常に大きい。例えば，UMACなどのMAC生成方法では暗号学的に衝突なしで一方向性を保証している安全なハッシュ関数が必要であり，ストリーム暗号として用いるには，擬似乱数器にさらに前述のハッシュ関数を実装する必要がある。，UMACは以下の非特許文献2で扱っている。

【０００８】

【非特許文献１】

V.Gligor, P.Donescu著「Lecture notes in Computer Science, vl.1796」Springer-Verlag出版，2000年，p.153-171

【非特許文献２】

Black, Halevi, Krawczyk, Krovetz, Rogaway, "UMAC: Fast and Secure Message Authentication," Advances in Cryptology, - CRYPTO' 99, Lecture Notes in Computer Science, Vol. 1666, Springer-Verlag, 1999

【０００９】

【発明が解決しようとする課題】

従来の暗号技術のほとんどは，復号化の際，改ざん検出をそれ自身ではすることができなかった。改ざん検出を行うとき，異なるふたつの鍵共有の必要性，メッセージの２倍となる乱数の必要性，独立の処理，別の暗号学的要素関数の追加実装などが必要であった。

【００１０】

処理速度の面についての課題として，ブロック暗号のこれまで知られた操作モードでは並列度や事前計算等の可能性がなく，高並列処理や高速処理には向いていないという点があり，ストリーム暗号のこれまで知られた操作モードでは，演算量が多いということや，必要な乱数が多いという理由で，ソフトウェア実装における処理速度がブロック暗号の場合と同程度であり，より高速な処理速度が必要とされている。

【００１１】

【課題を解決するための手段】

本発明は効率的で証明可能安全な暗号方法であり，改ざん検出も復号化と同時に可能であり，またデータ秘匿，データ改ざんに対する安全性については，証明可能なメッセージ認証付き暗号方法とその装置を提供する。

【００１２】

本発明は擬似乱数生成器の高速処理性能を生かしながら，事前計算や並列処理の利点をもつ，共通鍵暗号の方法とその装置を提供する。

【００１３】

本発明は従来のブロック暗号よりも高速な処理が可能というだけではなく，シングルパスで実装が可能でソフトウェアにおいて極めて効率的な処理が可能な暗号方法とその装置を提供する。

【００１４】

本発明は小さなプログラムで実装可能なストリーム暗号方法とその装置を提供する。

【００１５】

本発明は，その一態様において，乱数を生成して暗号化と認証処理を行い，事前計算と並列計算を達成する。また，生成する乱数の長さは，メッセージ長Nに対して2Nより少ない乱数を用いて，暗号処理と認証処理を行う。

【００１６】

具体的には，疑似乱数生成器を使って乱数を生成しそれらをブロック毎に分割する。また平文もブロック毎に分割する。乱数ブロックと平文ブロックを排他的論理和し暗号文ブロックを得る。非特許文献２で扱っているハッシュ関数NHは，乱数ブロックを鍵入力とし，生成された暗号文のメッセージ認証子を生成する。乱数生成は事前計算可能であり，暗号文ブロック生成演算は並列処理可能であり，ハッシュ関数NHも並列処理可能であるため，高速計算ができる。

【００１７】

【発明の実施の形態】

(第一の実施形態)

以下，本発明の第一の実施形態を図を用いて説明する。なお、ビットごとの排他的論理和は、以下の説明ではEORと表し、各図においてはプラス記号を円で囲んだ記号で表す。

(第一の実施形態)

図1はネットワーク1001によって接続されたコンピュータA1002，コンピュータB1003を含む，コンピュータA1002からコンピュータB1003への暗号通信を目的としたシステム構成を示すものである。コンピュータA1002は内部に演算装置(以下CPUという)1004，記憶装置(揮発性，不揮発性を問わない。以下RAMという)1005，ネットワークインターフェース1006を装備しており，外部にはコンピュータA1002をユーザが操作するためのディスプレイ1007とキーボード1008が接続されている。RAM1005には暗号化処理プログラムPROG1_1009，乱数生成処理プログラムPROG2_1010，コンピュータA1002とコンピュータB1003間のみで共有されている秘密情報である秘密鍵K1011，コンピュータA1002とコンピュータB1003間で共有さ

れているデータである初期ベクトルI1013，それに暗号化してコンピュータB1003に送信したいデータであるメッセージM1014が保存されている。コンピュータB1003は内部にCPU1015，RAM1016，ネットワークインターフェース1017を装備しており，外部にはコンピュータB1003をユーザが操作するためのディスプレイ1018とキーボード1019が接続されている。RAM1016には復号化処理プログラムPROG3_1020，乱数生成処理プログラムPROG2_1021，秘密鍵K1011が保存されている。

【００１８】

コンピュータA1002は，暗号化処理プログラムPROG1_1009を実行し，メッセージM1014の暗号文C1022を作成し，ネットワークインターフェース1006を通してネットワーク1001へ送信する。コンピュータB1003は，ネットワークインターフェース1017を通して受信したあと，復号化処理プログラムPROG3_1020を実行し，もし改ざんが検出されなければ復号化結果をRAM1016に保存する。

【００１９】

各プログラムは，互いのコンピュータまたは他のコンピュータから通信媒体、すなわちネットワーク1001またはネットワーク1001上を伝搬する搬送波、を介して，またはCD，FDなど可搬型記憶媒体を介してRAMに導入することができる。各プログラムは，各コンピュータのオペレーティングシステム(図示していない)の元で動作するように構成することも可能である。また、各プログラムによる処理は、CPUが当該プログラムをメモリから読み出して実行することにより、各コンピュータ上で実現される。

【００２０】

暗号化処理プログラムPROG1_1009は，コンピュータA1002において，RAM1005から読み出されて，CPU1004により実行される。暗号化処理プログラムPROG1_1009は，サブルーチンとして乱数生成処理PROG2_1010を内部で呼び出し，入力秘密鍵K1011，メッセージM1014に対して，暗号文C1022を出力する。

【００２１】

復号化処理プログラムPROG3_1020は，コンピュータB1003において，RAM1016から読み出されてCPU1015により実行される。復号化処理プログラムPROG3_1020は，サブルーチンとして乱数生成処理PROG2_1021を内部で呼び出し，入力秘密鍵K1

011，暗号文C1022に対して，メッセージ，または改ざん検知警告を出力する。

【００２２】

暗号化処理プログラムPROG1_1009の処理の流れを説明する。

ステップ2002：データセットサブルーチン。秘密鍵Kの入力を待つ。

ステップ2003：平文準備サブルーチン。平文の入力を待ち，平文が与えられたあと決められたパディングを行い，最後に64ビットごとに区切って平文ブロックの列P_i($1 \leqq i \leqq N$)を出力する。但し，Nは偶数とする。

ステップ2004：乱数生成サブルーチン。秘密鍵Kと初期ベクトルIから擬似乱数列R_i($1 \leqq i \leqq N+1$)を出力する。

ステップ2005：暗号化サブルーチン。擬似乱数列R_iと平文ブロック列P_i($1 \leqq i \leqq N$)を使って，暗号文ブロックC_i($1 \leqq i \leqq N+2$)を出力する。

ステップ2006：ステップ2005で得られた暗号文ブロックC_i($1 \leqq i \leqq N+2$)を順にビット連結し，暗号文Cとして出力する。

【００２３】

平文準備サブルーチンの処理を図2を用いて説明する。

ステップ2202：暗号処理に用いるメッセージMの入力を待つ。メッセージMは，キーボード1008から入力されたり，RAM内に保存されていたり，他の記憶媒体から導入されたりする。

ステップ2203：メッセージの長さを示すデータをパディング。メッセージMの先頭にメッセージMのビット長を表す64ビット二進数データを付加する。

ステップ2204：メッセージ長をそろえるパディング。後の暗号処理のためパディング後のデータを128ビットの整数倍にする。メッセージMの長さをLビットとすると$128-(L \pmod{128})$個の0をステップ2203で長さデータを付加されたメッセージの末尾にパディングする。

ステップ2206：メッセージデータの平文ブロックへ分割。ステップ2205の結果得られたデータを64ビットのブロックに区切り順にP_1，P_2，...，P_Nとする。

【００２４】

乱数生成サブルーチンの処理を図3を用いて説明する。

ステップ2302：必要パラメータの入力。パディング後のメッセージブロック数N

と，　初期ベクトルI

秘密鍵Kを得る。

ステップ2303：擬似乱数列Rの生成。乱数生成処理プログラムPROG2を呼び出し，長さ64(N＋1)ビットの擬似乱数列を生成，出力をRとする。

ステップ2304：乱数列Rをブロックに分割。擬似乱数列Rを64ビットごとに区切り順にR_1，R_2，...，R_{N+1}とする。

【００２５】

暗号化，メッセージ認証子生成セットアップのサブルーチンの処理を図4を用いて説明する。

ステップ2403：カウンタ初期化。i=1とする。

ステップ2404：　暗号文ブロックC_i計算。$C_i \leftarrow M_i$ EOR R_i とする。

ステップ2406：i=Nならばステップ2408を実行。

ステップ2407：カウンタiをインクリメントしステップ2404へ戻る。

ステップ2408：$C_i(1 \leqq i \leqq N)$を順にビット連結し，Sとする。$R_i(2 \leqq i \leqq N+1)$を順にビット連結し，Rとする。

ステップ2409：$NH_R(S)$の出力を64ビットごとに区切り，C_{N+1}，C_{N+2}とする。

【００２６】

ハッシュ関数$NH_R()$については，図11で解説する。

【００２７】

復号化処理プログラムPROG3_1020の処理の流れを図5を用いて説明する。

ステップ2502：データセットサブルーチン。秘密鍵Kの入力を待つ。

ステップ2503：暗号文準備サブルーチン。暗号文C’の入力を待ち，暗号文C’が与えられたあと64ビットごとに区切って暗号文ブロックの列$C'_i(1 \leqq i \leqq N+2)$を出力する。

ステップ2504：乱数生成サブルーチン。秘密鍵Kから擬似乱数列$R_i(1 \leqq i \leqq N+1)$を出力する。

ステップ2505：$C_i(1 \leqq i \leqq N)$を順にビット連結し，Sとする。$R_i(2 \leqq i \leqq N+1)$を順にビット連結し，Rとして$NH_R(S)$を計算する。

ステップ2506：　$NH_R(S) = C'_{N+1} || C'_{N+2}$ならばステップ2508に進む。そうで

ないならばステップ2507に進む。

ステップ2507：拒否(非受理)を出力。ステップ2511に進む。

ステップ2508：復号化サブルーチン。擬似乱数列R_i(1≦i≦N)，暗号文ブロック列C’$_i$(1≦i≦N)を使って，平文ブロックP’$_i$(1≦i≦N)を出力する。

ステップ2509：平文切り出しサブルーチン。平文ブロックの列P’$_i$をデータ列，L’，M’に分割する。

ステップ2510：M’をRAMへ格納する。

ステップ2511では，復号化処理プログラムは，結果(受理/非受理あるいは復号結果)をディスプレイ1018に出力して，ユーザに結果を通知する。

【００２８】

暗号文準備サブルーチンの処理を図6を用いて説明する。

ステップ2602：暗号文C’の入力を待つ。

ステップ2603：暗号文C’を64ビットごとに区切り，順にC’$_1$，C’$_2$，...，C’$_{N+1}$，C’$_{N+2}$とする。

【００２９】

復号化サブルーチンの処理を図7を用いて説明する。

ステップ2703：カウンタ初期化。i=1とする。

ステップ2704：平文ブロックP’$_i$計算。P’$_i$=C’$_i$^R_iとする。

ステップ2706：i=Nでないならばステップ2707を実行。

ステップ2707：カウンタiをインクリメントしステップ2704へ戻る。

【００３０】

平文切り出しサブルーチンの処理を図8を用いて説明する。

ステップ2802：L’を最初の64ビットの平文ブロックとする。

ステップ2803：復号文ブロックのうちP’$_2$の最上位ビットからL’ビットのデータまでをM’とする。

【００３１】

図9は，暗号化処理の説明図である。

【００３２】

メッセージM2931に長さ2930，適当なパディング2932をそれぞれ付加し，平文P

2934を生成する。

【００３３】

これを64ビットに分割したものをそれぞれP_1_2935，P_2_2936，...，P_N_2938とする。

【００３４】

P_1_2935はR_1_2920と排他論理和を取り，暗号文ブロックC_1_2943を得る。

【００３５】

P_2_2936はR_2_2921と排他論理和を取り，暗号文ブロックC_2_2944を得る。

【００３６】

同様にしてP_N_2938まで処理を行い，暗号文ブロックC_1_2943，C_2_2944，...，C_N_2947を得る。R_2_2921，R_3_2921，R_{N+1}_2928をこの順で連結したものRとC_1_2943，C_2_2944，...，C_N_2947をこの順で連結したものSを入力として$NH_R(S)$を計算する。

$NH_R(S)$の出力をC_{N+1}_2948，C_{N+2}_2949とブロック分割し，C_1_2943，C_2_2944，...，C_N_2947，C_{N+1}_2948，C_{N+2}_2949この順で連結し，暗号文C_2956を得る。

【００３７】

図10は，復号化処理の説明図である。

【００３８】

暗号文C'_4030を64ビットのブロックに分割し，C'_1_4035，C'_2_4036，...，C'_N_7037，C'_{N+1}_4038，C'_{N+2}_4039とする。

R_2_4021，R_3_4022，...，R_{N+1}_4028をこの順で連結したものRとして，C'_1_4035，C'_2_4036，...，C'_N_7037をこの順で連結したものSを入力として$NH_R(S)$を計算し，$NH_R(S)=C_{N+2}$'_4038||C_{N+1}'_4039ならば，次に進む。

【００３９】

C_1'_4035はR_1_4020と排他論理和を取り，平文ブロックP_1'_4043を得る。

【００４０】

C_2'_4036はR_2_4021と排他論理和を取り，平文ブロックP_2'_4044を得る。

【００４１】

同様にしてC_N'_4037まで処理を行い，平文ブロックP_1'_4043，P_2'_4044，

..., P_N'_4047を得た後，これらをこの順に連結し，平文P'_4050とする。これをL'_4051，M'_4052に分割する。

【００４２】

図11を用いて，非特許文献2で扱っているハッシュ関数$NH_R(S)$の解説を行う。

【００４３】

この関数はメッセージMと鍵Kを入力とし，メッセージ認証子Cを生成し，出力する。は次のようにして行う。ただし，以下のアルゴリズムで，
矢印←はデータの代入を，||は結合を，それぞれ表す。$M=M_1||\cdots||M_N$，$K=K_1||\cdots||K_N$とする。

【００４４】

【数１】

$$H_i \leftarrow M_i + K_i \ (1 \leq i \leq N)$$
$$S_i \leftarrow H_{2i-1} \times H_{2i} \ (1 \leq i \leq N/2)$$
$$C \leftarrow S_1 + S_2 \ldots + S_{N/2}$$

【００４５】

最後にメッセージ認証子Cを出力する。

【００４６】

第一の実施形態において，暗号処理とメッセージ認証子生成という二つの処理に必要な擬似乱数の長さはメッセージのそれとほぼ同じで十分である。

【００４７】

また、一般的なCPUを使用した計算機上で，本実施形態による疑似乱数生成器はブロック暗号の中で最も高速なAESに比べ2倍以上高速な処理が可能である。従って，従来技術であるiaPCBCモードに比べ、同一環境上で2倍以上高速な処理が可能である。

(第二の実施形態)

以下，本発明の第二の実施形態について説明する。基本的には，第一の実施形態と同じだが，変更点だけ以下に示す。

【００４８】

暗号化処理プログラムPROG1_1009の処理の流れを説明する。

ステップ5002：データセットサブルーチン。秘密鍵Kの入力を待つ。

ステップ5003：平文準備サブルーチン。平文の入力を待ち，平文が与えられたあと決められたパディングを行い，最後に64ビットごとに区切って平文ブロックの列P_i(1≦i≦N)を出力する。但し，Nは偶数とする。

ステップ5004：乱数生成サブルーチン。秘密鍵Kと初期ベクトルIから64(3N/2＋1)ビットの擬似乱数列を出力する。

ステップ5005：暗号化サブルーチン。ステップ5004で得られた擬似乱数列と平文ブロック列P_i(1≦i≦N)を使って，暗号文ブロックC_i(1≦i≦N＋2)を出力する。

ステップ5006：ステップ5005で得られた暗号文ブロックC_i(1≦i≦N＋2)を順にビット連結し，暗号文Cとして出力する。

【００４９】

乱数生成サブルーチンの処理を図12を用いて説明する。

ステップ5302：必要パラメータの入力。パディング後のメッセージブロック数Nと，　初期ベクトルIと、秘密鍵Kを得る。

ステップ5303：擬似乱数列Rの生成。乱数生成処理プログラムPROG2を呼び出し，長さ

64(3N/2＋1)ビットの擬似乱数列Rを生成する。

ステップ5304：乱数列Rをブロックに分割。擬似乱数列Rを64ビットごとに区切り順にR_1，R_2，...，R_{N+1}...，$R_{3N/2+1}$とする。

ステップ5305：R_{N+1}...，$R_{3N/2}$をこの順に連結し，R’とする。

ステップ5306：R_{N+2}...，$R_{3N/2+1}$をこの順に連結し，R’’とする。

【００５０】

暗号化，メッセージ認証子生成セットアップのサブルーチンの処理を図13を用いて説明する。

ステップ5403：カウンタ初期化。i=1とする。

ステップ5404：　暗号文ブロックC_i計算。$C_i \leftarrow M_i$ EOR R_i とする。

ステップ5405：i=Nならばステップ5407を実行。

ステップ5406：カウンタiをインクリメントしステップ5404へ戻る。

ステップ5407：カウンタ初期化。i=1とする。

ステップ5408：C_iを32ビットごとに区切り，$C_{i,H}$，$C_{i,L}$とする。

ステップ5409：i=N/2ならばステップ5411を実行。

ステップ5410：カウンタiをインクリメントしステップ5408へ戻る。

ステップ5411：$C_{1,H}$，$C_{1,L}$，…，$C_{N/2,H}$，$C_{N/2,L}$を順にビット連結し，Sとする。

ステップ5412：$NH_R'(S)$の出力をC_{N+1}とする。

ステップ5413：$NH_R''(S)$の出力をC_{N+2}とする。

【００５１】

復号化処理プログラムの処理の流れを図14を用いて説明する。

ステップ5502：データセットサブルーチン。秘密鍵Kの入力を待つ。

ステップ5503：暗号文準備サブルーチン。暗号文C’の入力を待ち，暗号文C’が与えられたあと64ビットごとに区切って暗号文ブロックの列$C'_i(1\leqq i\leqq N+2)$を出力する。

ステップ5504：乱数生成サブルーチン。秘密鍵Kから擬似乱数列$R_i(1\leqq i\leqq 3N/2+1)$，R’，R’’を出力する。

ステップ5505：$C_i(1\leqq i\leqq N)$を順にビット連結し，Sとする。$NH_R'(S)$と$NH_R''(S)$を計算する。

ステップ5506：$NH_R'(S)=C'_{N+1}$かつ$NH_R''(S)=C'_{N+2}$ならばステップ5508に進む。そうでないならばステップ5507に進む。

ステップ5507：拒否(非受理)を出力。ステップ5511に進む。

ステップ5508：復号化サブルーチン。擬似乱数列R_i，暗号文ブロック列$C'_i(1\leqq i\leqq N)$を使って，平文ブロック$P'_i(1\leqq i\leqq N)$を出力する。

ステップ5509：平文切り出しサブルーチン。平文ブロックの列P'_iをデータ列L’，M’に分割する。

ステップ5510：M’をRAMへ格納する。

ステップ5510では，復号化処理プログラムは，結果(受理/非受理あるいは復号結果)をディスプレイ10018に出力して，ユーザに結果を通知する。

【００５２】

図15は，暗号化処理の説明図である。

【００５３】

メッセージM5931に長さ5930，適当なパディング5932をそれぞれ付加し，平文P5934を生成する。これを64ビットに分割したものをそれぞれP_1_5935，P_2_5936，$P_{N/2}$_5937，...，P_N_5938とする。P_1_5935はR_1_5920と排他論理和を取り，暗号文ブロックC_1_5943を得る。P_2_5936はR_2_5921と排他論理和を取り，暗号文ブロックC_2_5944を得る。

【００５４】

同様にしてP_N_5938まで処理を行い，暗号文ブロックC_1_5943，C_2_5944，...，C_N_5947を得る。C_1_5943，C_2_5944，...，$C_{N/2}$_5945をこの順で連結したものSを入力としてNH_R'(S)を計算し，その出力をC_{N+1}_5948とする。

【００５５】

NH_R''(S)を計算し，その出力をC_{N+2}_5949とする。C_1_5943，C_2_5944，...，$C_{N/2}$_5945，...，C_N_5947，C_{N+1}_5948，C_{N+2}_5949この順で連結し，暗号文C_5956を得る。

【００５６】

図16は，復号化処理の説明図である。

暗号文C'_6030を64ビットのブロックに分割し，C'$_1$_6035，C'$_2$_6036，...，C'$_N$_7037，C'$_{N+1}$_6038，C'$_{N+2}$_6039とする。C'$_1$_6033，C'$_2$_6034，C'$_{N/2}$_7035，...，C'$_N$_7037をこの順で連結したものSを入力としてNH_R(S)を計算し，NH_R'(S)＝C_{N+1}'_6038かつ，NH_R''(S)＝C_{N+2}'_6039ならば，次に進む。

【００５７】

C_1'_6033はR_1_6020と排他論理和を取り，平文ブロックP_1'_6043を得る。C_2'_6034はR_2_6031と排他論理和を取り，平文ブロックP_2'_6044を得る。

【００５８】

同様にしてC_N'_6037まで処理を行い，平文ブロックP_1'_6043，P_2'_6044，...，P_N'_6047を得た後，これらをこの順に連結し，平文P'_6050とする。これをL'_6051，M'_6052に分割する。

【００５９】

第二の実施形態において，暗号処理とメッセージ認証子生成という二つの処理に必要な擬似乱数の長さはメッセージのそれのほぼ1.5倍である。また，一般的なCPUを使用した計算機上で，本実施形態による疑似乱数生成器はブロック暗号の中で最も高速なAESに比べ2倍以上高速に乱数生成処理が可能である。以上の考察から，第二の実施形態の方法は、従来技術であるiaPCBCモードに比べ、同一環境上で、4/3倍以上高速な処理が可能である。

【００６０】

また、非特許文献2の定理2を，w = 32 ，t = 2として第二の実施形態に適用することにより，安全性証明ができる。すなわち，長さが同じである二つの異なるメッセージに対し，それらのメッセージ認証子が等しくなる確率は2^{-64}である。

【００６１】

【発明の効果】

本発明によれば、メッセージ認証付き暗号方法をソフトウェアで実装する際，処理速度を高速化することができる。

【図面の簡単な説明】

【図１】

各実施形態のシステム構成図を示す。

【図２】

平文準備サブルーチンのフロー図を示す。

【図３】

乱数生成サブルーチンのフロー図を示す。

【図４】

暗号化サブルーチンのフロー図を示す。

【図５】

図1の復号化処理プログラムのフロー図を示す。

【図６】

暗号文準備サブルーチンのフロー図を示す。

【図７】

復号化サブルーチンのフロー図を示す。

【図８】

平文切出しサブルーチンのフロー図を示す。

【図９】

暗号化処理の，データブロックによる図を示す。

【図１０】

復号化処理の，データブロックによる図を示す。

【図１１】

ハッシュ関数NHのフロー図を示す。

【図１２】

第二の実施形態の乱数生成2サブルーチンのフロー図を示す。

【図１３】

第二の実施形態の暗号化2サブルーチンのフロー図を示す。

【図１４】

第二の実施形態の復号化処理プログラムのフロー図を示す。

【図１５】

第二の実施形態の暗号化処理の，データブロックによる図を示す。

【図１６】

第二の実施形態の復号化処理の，データブロックによる図を示す。

【図１７】

第一の実施形態の暗号化処理と認証処理における，乱数共有方法の概念図を示す。

【符号の説明】

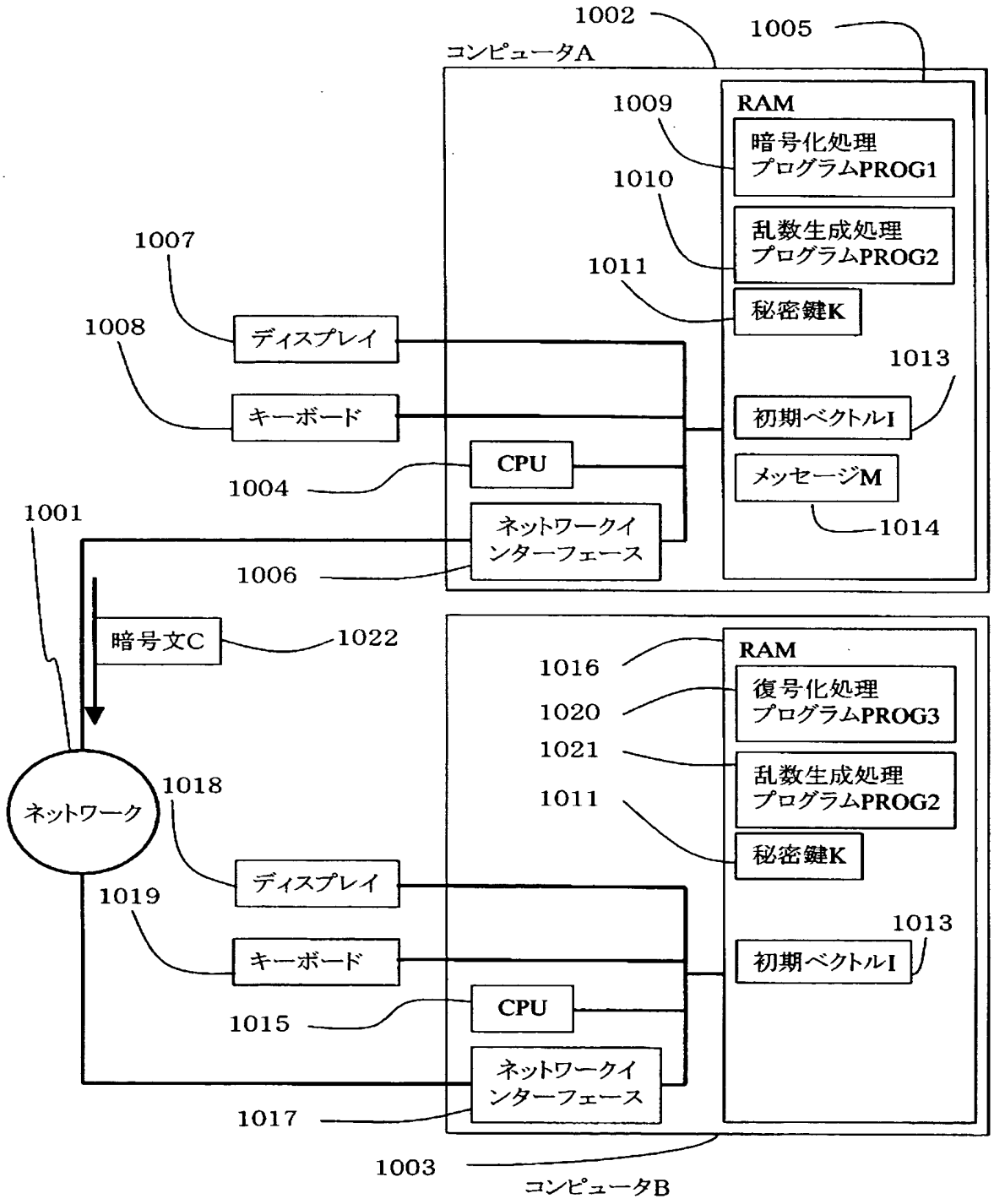
1001…ネットワーク、1002…コンピュータA、1003…コンピュータB、1004…CPU、1005…RAM、1006…ネットワークインターフェース、1007…ディスプレイ、1008…キーボード、1009…暗号化処理プログラムPROG1_、1010…乱数生成処理プログラムPROG2_、1011…秘密鍵K、1013…初期ベクトルI、1014…メッセージM、1015…CPU、1016…RAM、1017…ネットワークインターフェース、1018…ディスプレイ、1019…キーボード、1020…復号化処理プログラムPROG3_、1021…乱数生成処

理プログラムPROG2_、1022…暗号文C。

【書類名】　図面

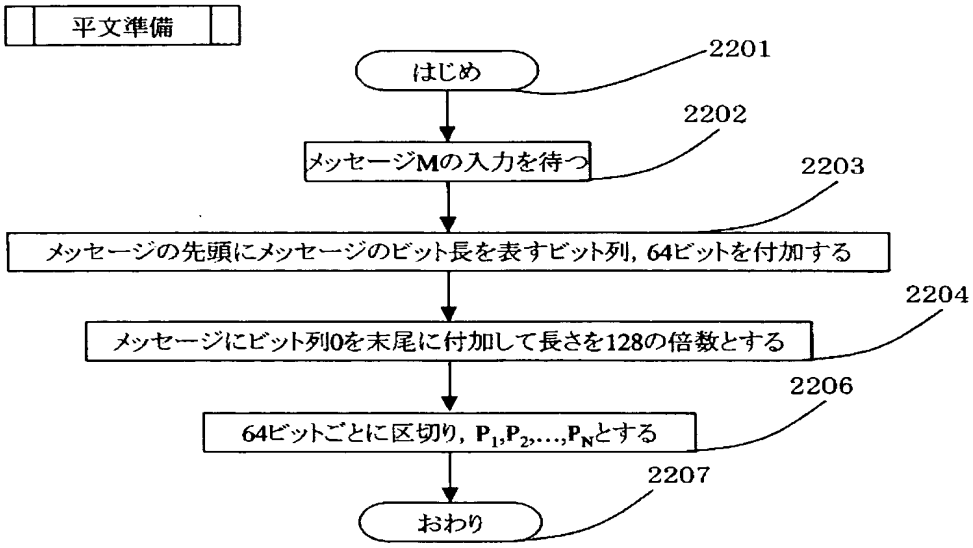
【図１】

図1


1002
1005
コンピュータA
RAM
1009
暗号化処理
プログラムPROG1
1010
乱数生成処理
プログラムPROG2
1011
秘密鍵K
1007
ディスプレイ
1008
1013
キーボード
初期ベクトルI
CPU
1004
メッセージM
1001
1014
ネットワークイ
ンターフェース
1006
暗号文C
1022
RAM
1016
復号化処理
プログラムPROG3
1020
1021
乱数生成処理
プログラムPROG2
1011
1018
ネットワーク
秘密鍵K
ディスプレイ
1019
1013
キーボード
初期ベクトルI
CPU
1015
ネットワークイ
ンターフェース
1017
1003
コンピュータB

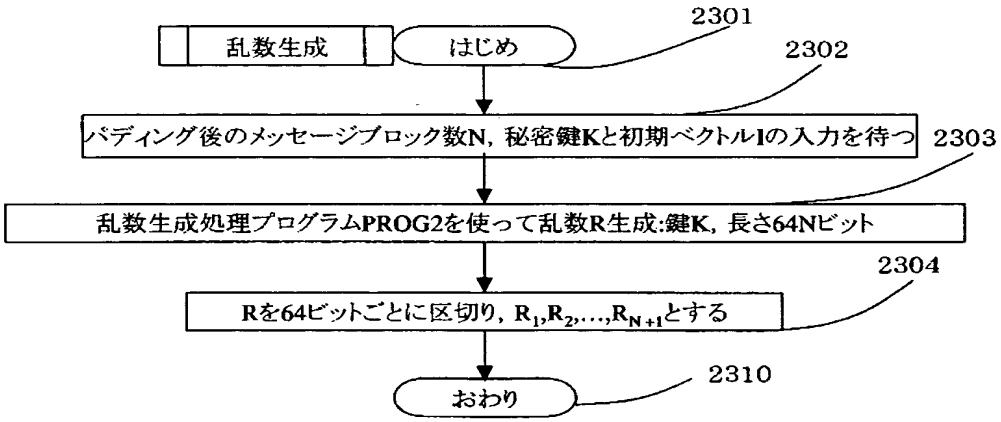
【図２】

図2


平文準備
はじめ
2201
メッセージMの入力を待つ
2202
メッセージの先頭にメッセージのビット長を表すビット列，64ビットを付加する
2203
メッセージにビット列0を末尾に付加して長さを128の倍数とする
2204
64ビットごとに区切り，$P_1,P_2,\ldots,P_N$とする
2206
おわり
2207

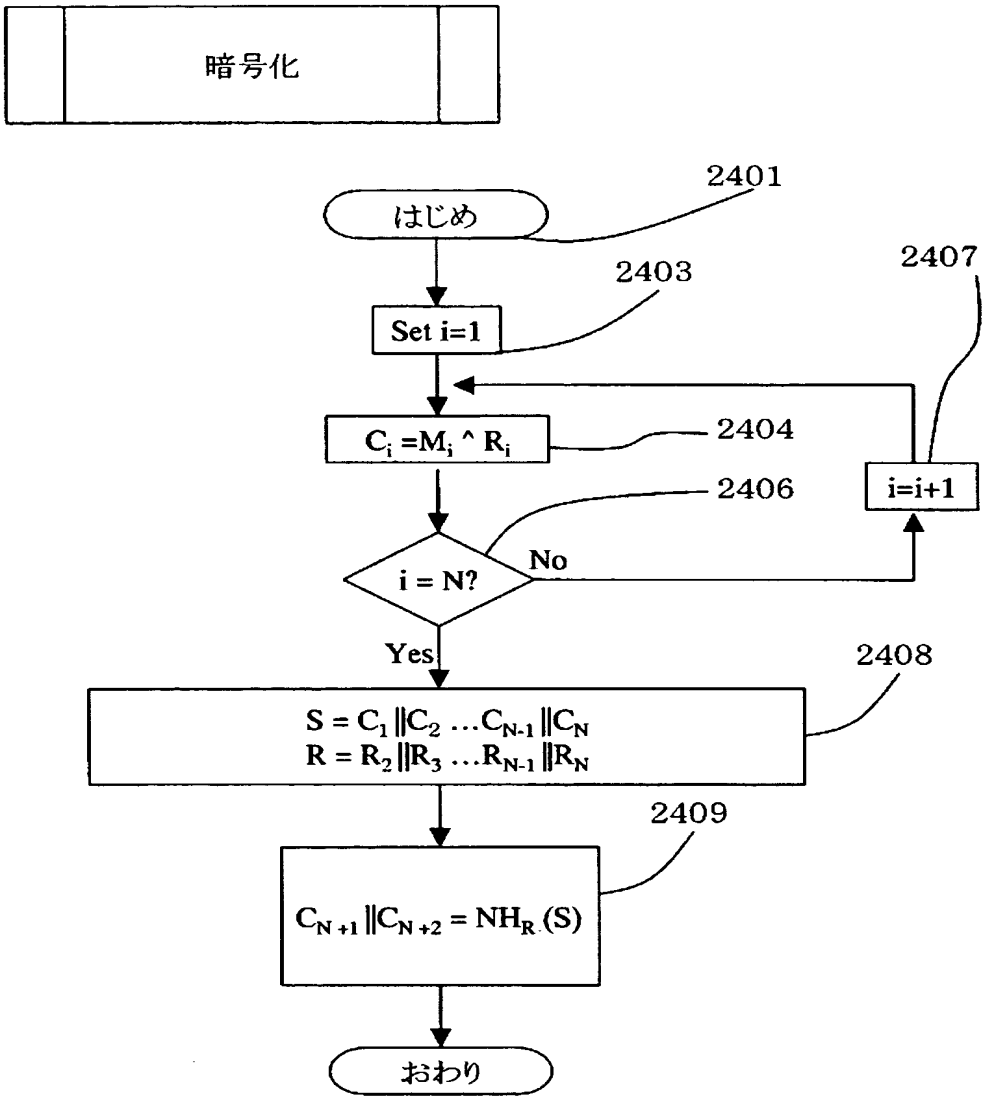
【図３】

図3


乱数生成
はじめ
2301
パディング後のメッセージブロック数N，秘密鍵Kと初期ベクトルIの入力を待つ
2302
乱数生成処理プログラムPROG2を使って乱数R生成:鍵K，長さ64Nビット
2303
Rを64ビットごとに区切り，$R_1,R_2,\ldots,R_{N+1}$とする
2304
おわり
2310

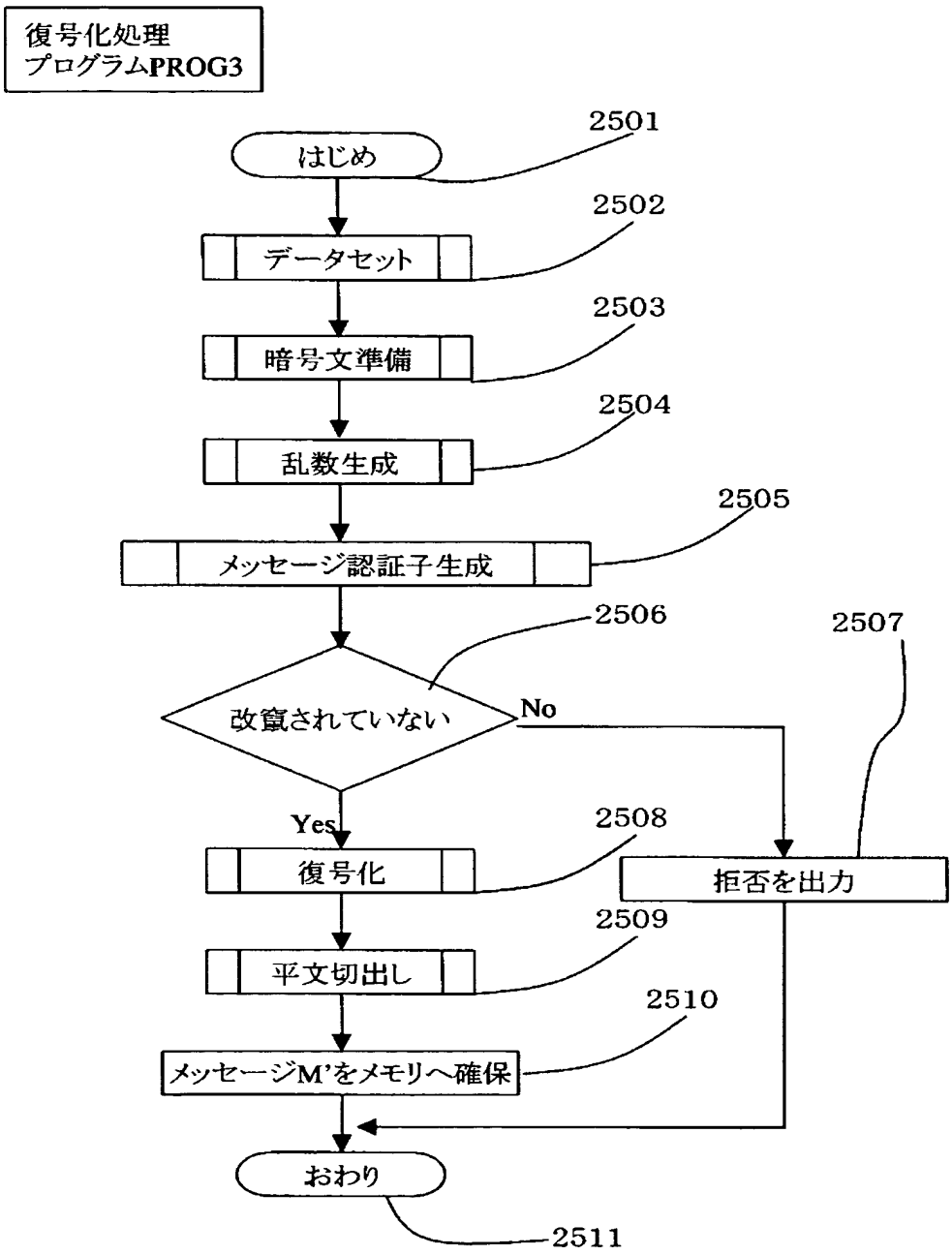
【図４】

図4


暗号化
2401
はじめ
2403
Set i=1
2407
$C_i = M_i \wedge R_i$
2404
i=i+1
2406
i = N?
No
Yes
2408
$S = C_1 \| C_2 \ldots C_{N-1} \| C_N$
$R = R_2 \| R_3 \ldots R_{N-1} \| R_N$
2409
$C_{N+1} \| C_{N+2} = NH_R(S)$
おわり

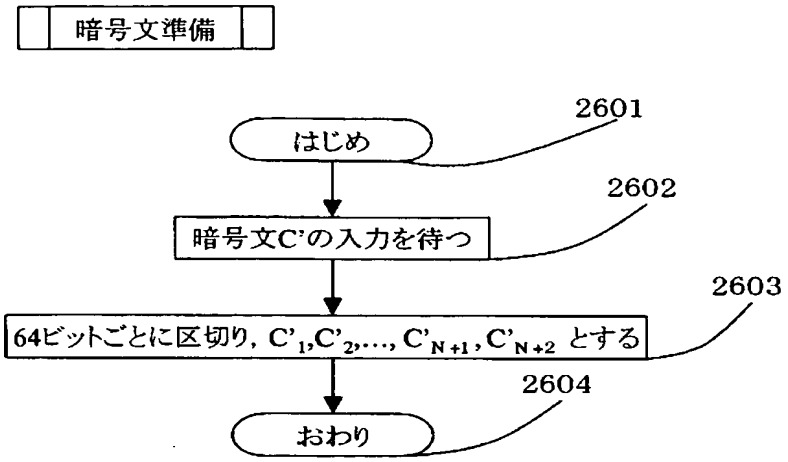
【図５】

図5


復号化処理
プログラムPROG3
はじめ
2501
データセット
2502
暗号文準備
2503
乱数生成
2504
メッセージ認証子生成
2505
改竄されていない
2506
No
2507
Yes
復号化
2508
拒否を出力
平文切出し
2509
メッセージM'をメモリへ確保
2510
おわり
2511

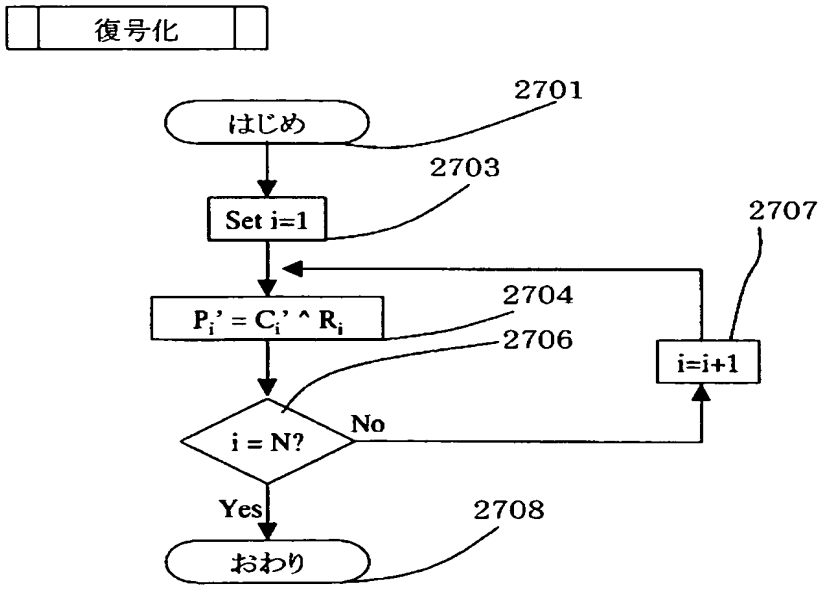
【図６】

図6


暗号文準備
2601
はじめ
2602
暗号文C'の入力を待つ
2603
64ビットごとに区切り，C'1,C'2,…, C'N+1, C'N+2 とする
2604
おわり

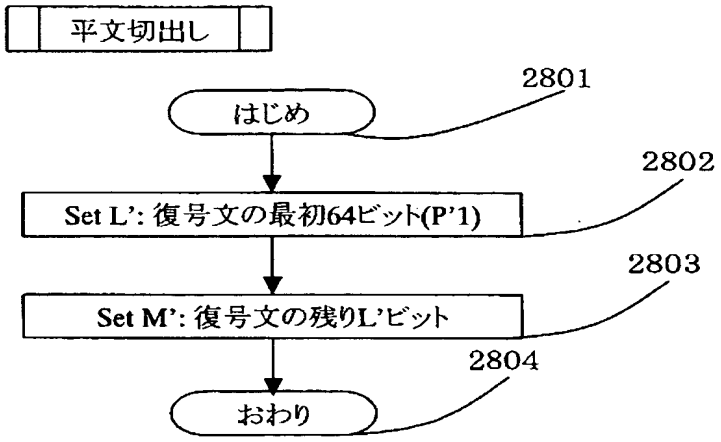
【図７】

図7


復号化
2701
はじめ
2703
Set i=1
2704
Pi' = Ci' ^ Ri
2706
i = N?
No
Yes
2707
i=i+1
2708
おわり

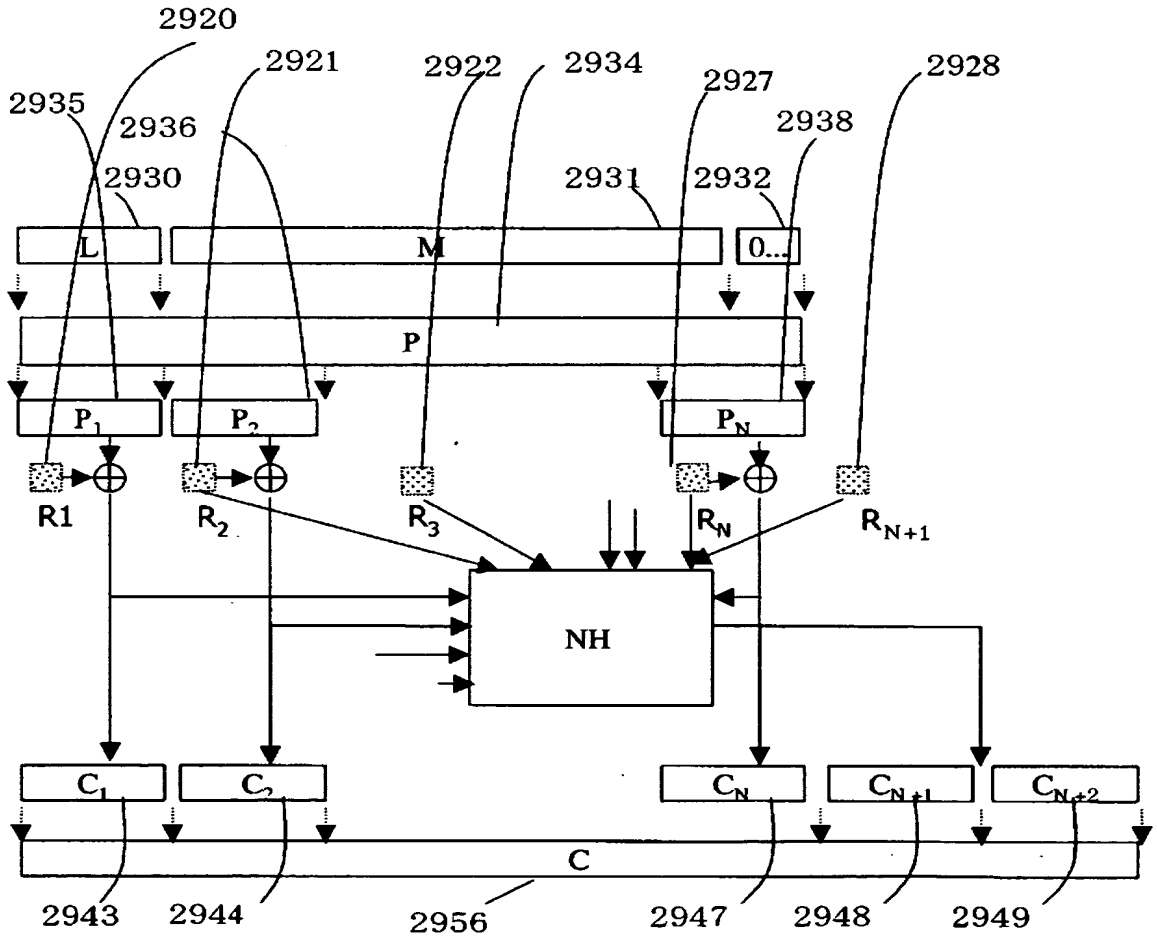
【図８】

図8


平文切出し
はじめ
2801
Set L': 復号文の最初64ビット(P'1)
2802
Set M': 復号文の残りL'ビット
2803
おわり
2804

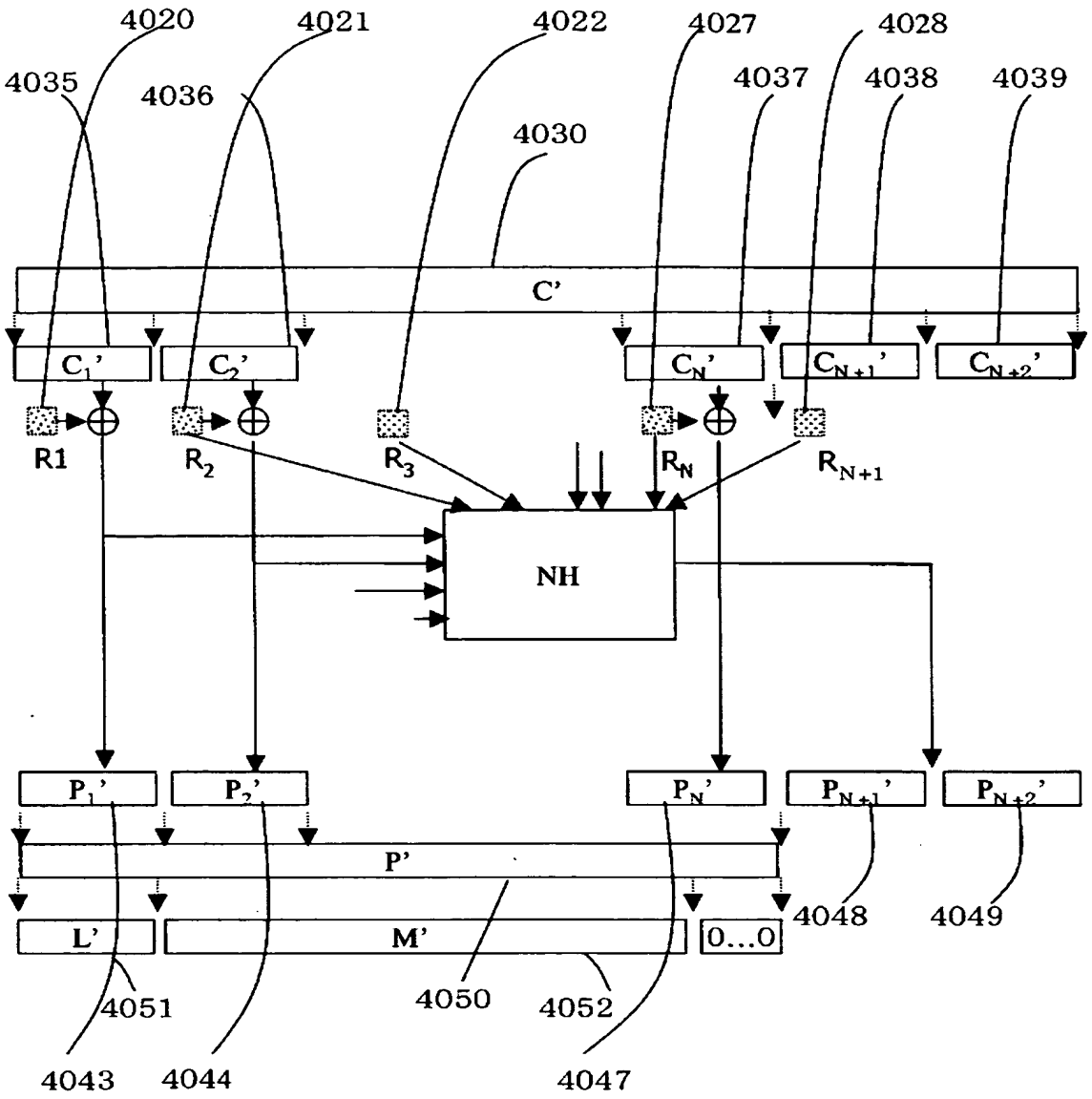
【図９】

図9


2920
2921
2922
2934
2927
2928
2935
2936
2938
2930
2931
2932
L
M
0...
P
P1
P2
PN
R1
R2
R3
RN
RN+1
NH
C1
C2
CN
CN+1
CN+2
C
2943
2944
2956
2947
2948
2949

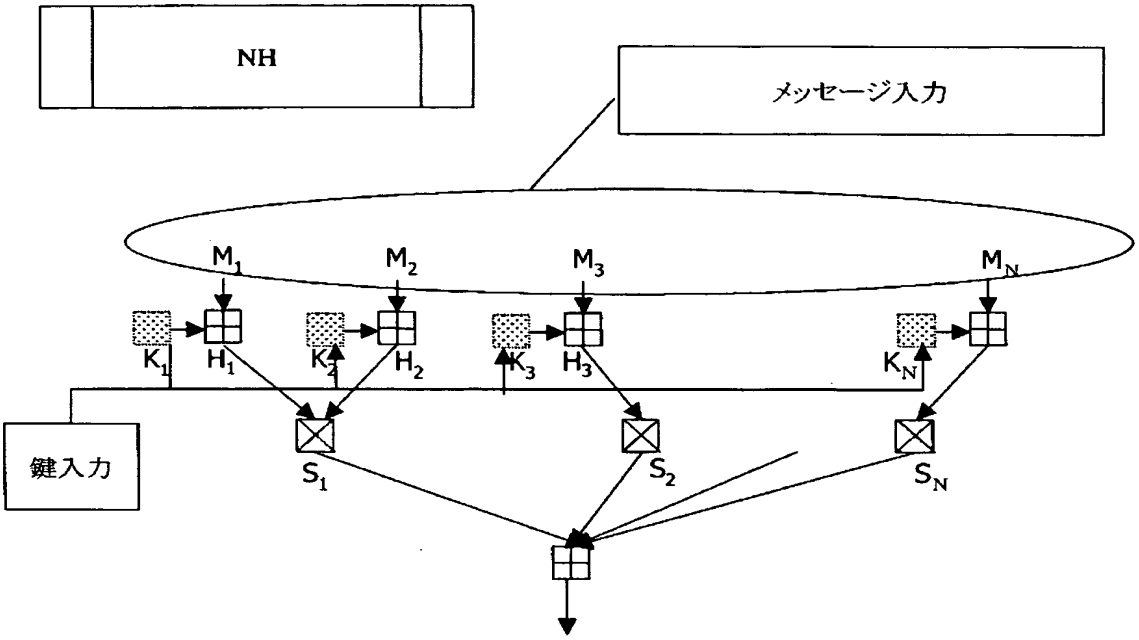
【図１０】

図10


4020
4021
4022
4027
4028
4035
4036
4037
4038
4039
4030
C'
C1'
C2'
CN'
CN+1'
CN+2'
R1
R2
R3
RN
RN+1
NH
P1'
P2'
PN'
PN+1'
PN+2'
P'
L'
M'
0...0
4048
4049
4051
4050
4052
4043
4044
4047

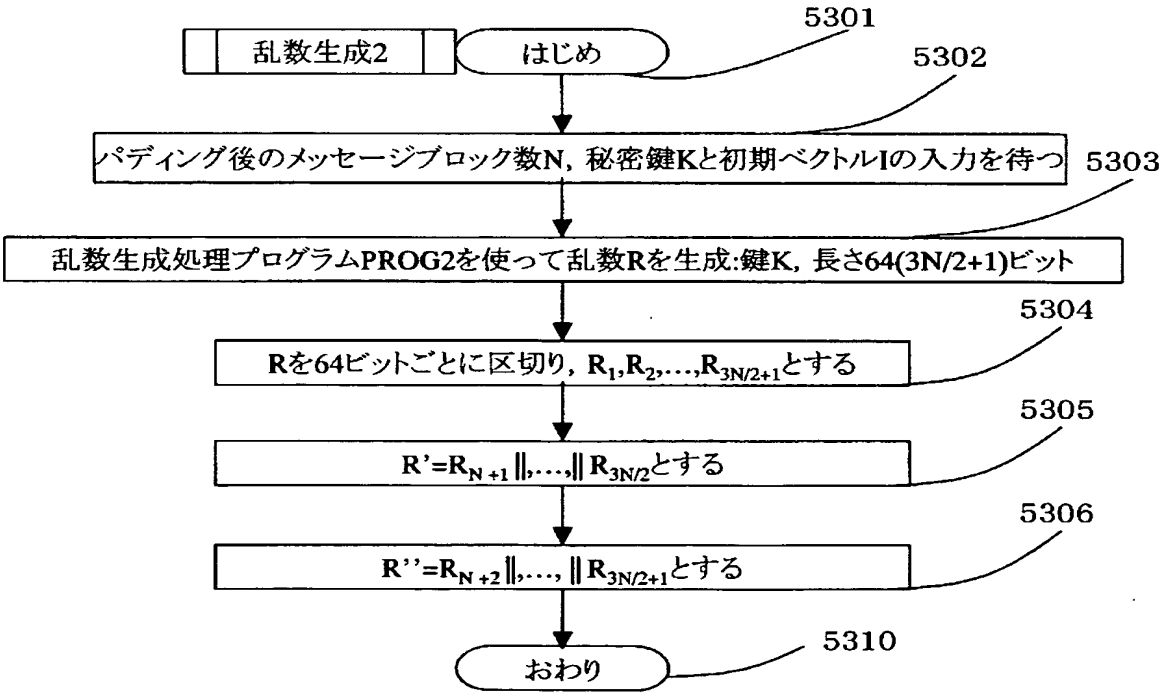
【図１１】

図11



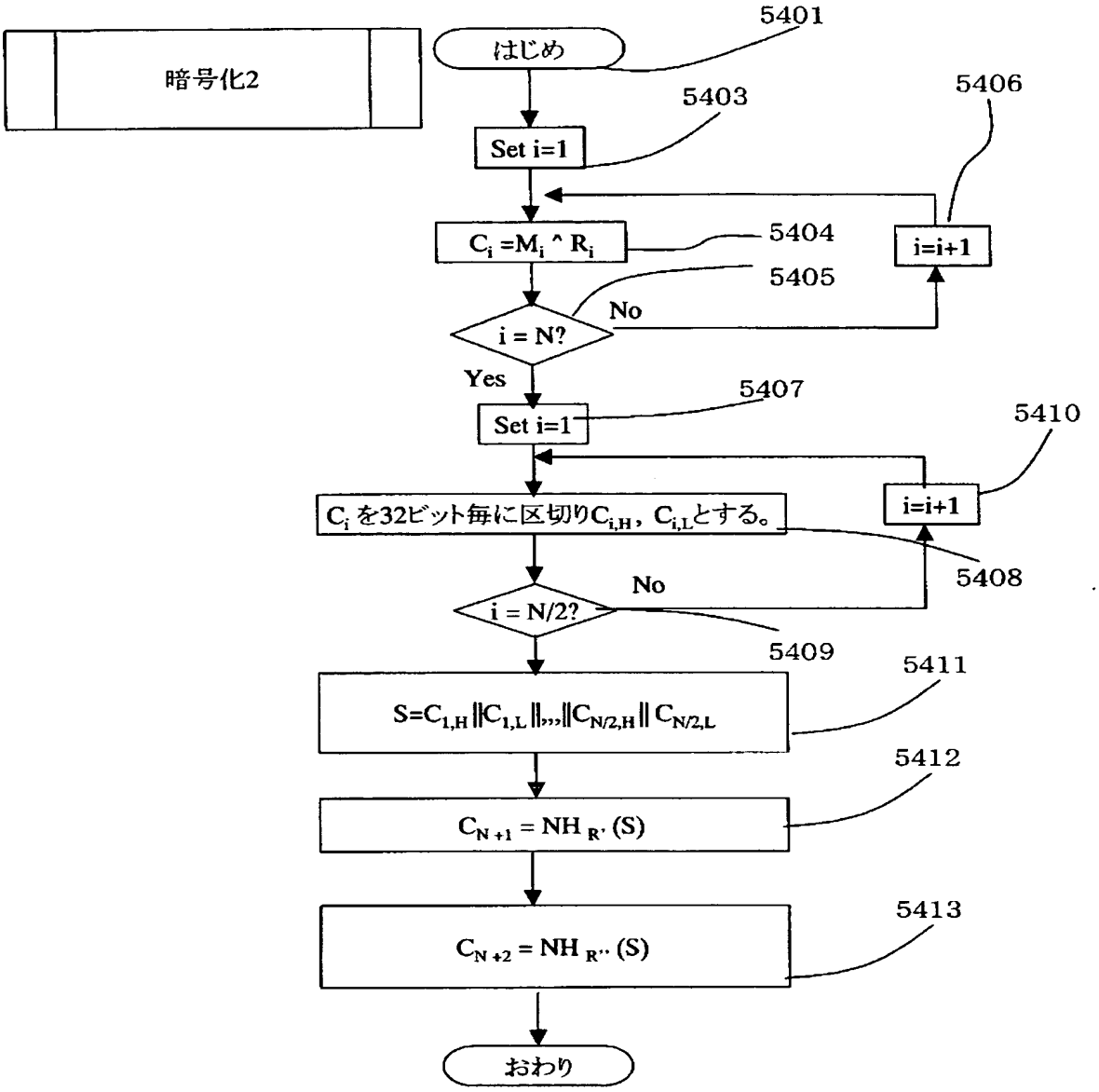
【図１２】

図12



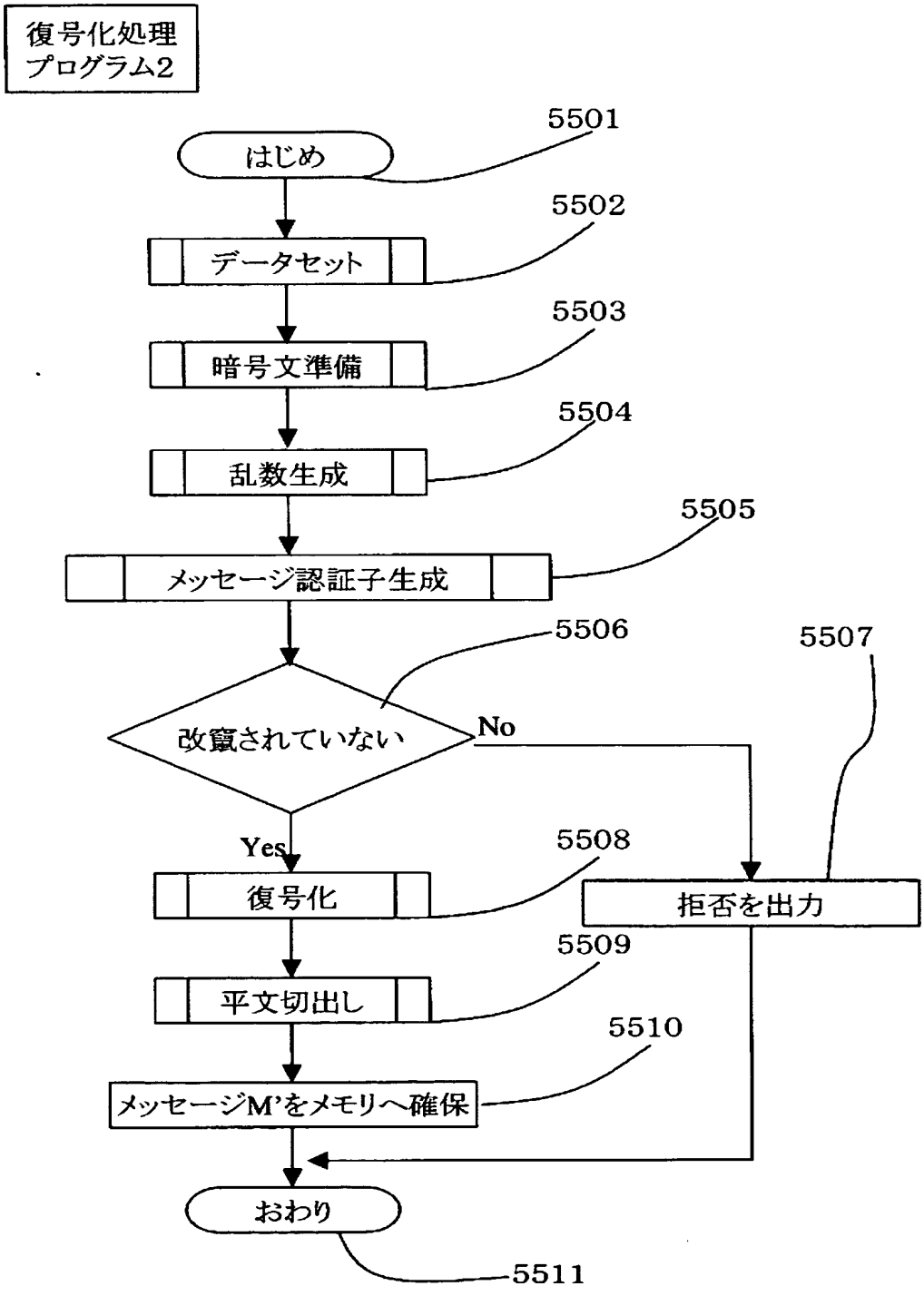
【図１３】

図13


暗号化2
はじめ
5401
Set i=1
5403
$C_i = M_i \wedge R_i$
5404
$i = N?$
5405
No
Yes
i=i+1
5406
Set i=1
5407
C_iを32ビット毎に区切り$C_{i,H}$, $C_{i,L}$とする。
i=i+1
5410
5408
$i = N/2?$
No
5409
$S=C_{1,H}\|C_{1,L}\|,,,\|C_{N/2,H}\|C_{N/2,L}$
5411
$C_{N+1} = NH_{R'}(S)$
5412
$C_{N+2} = NH_{R''}(S)$
5413
おわり

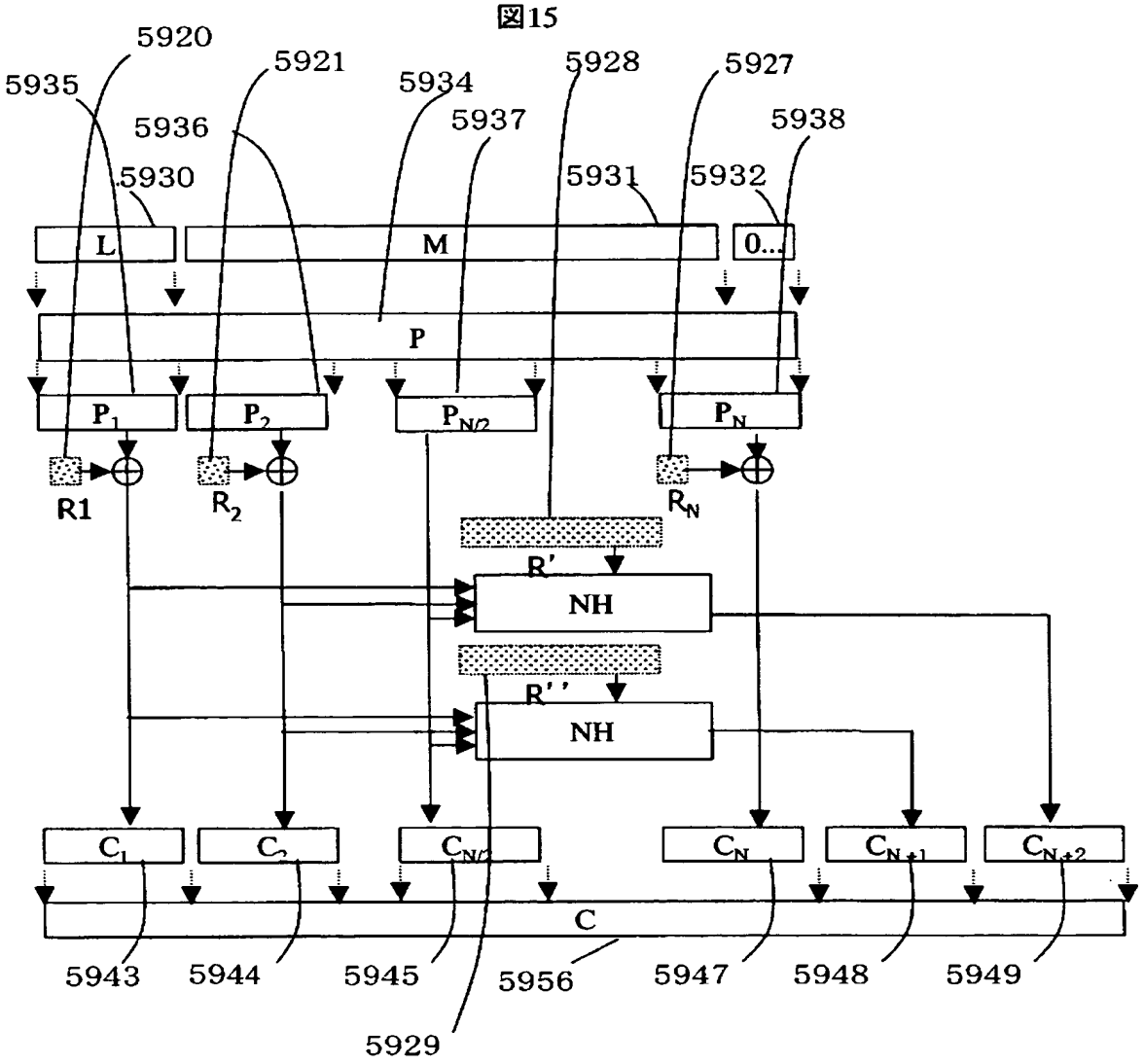
【図１４】

図14


復号化処理
プログラム2
5501
はじめ
5502
データセット
5503
暗号文準備
5504
乱数生成
5505
メッセージ認証子生成
5506
改竄されていない
No
5507
Yes
5508
復号化
拒否を出力
5509
平文切出し
5510
メッセージM'をメモリへ確保
おわり
5511

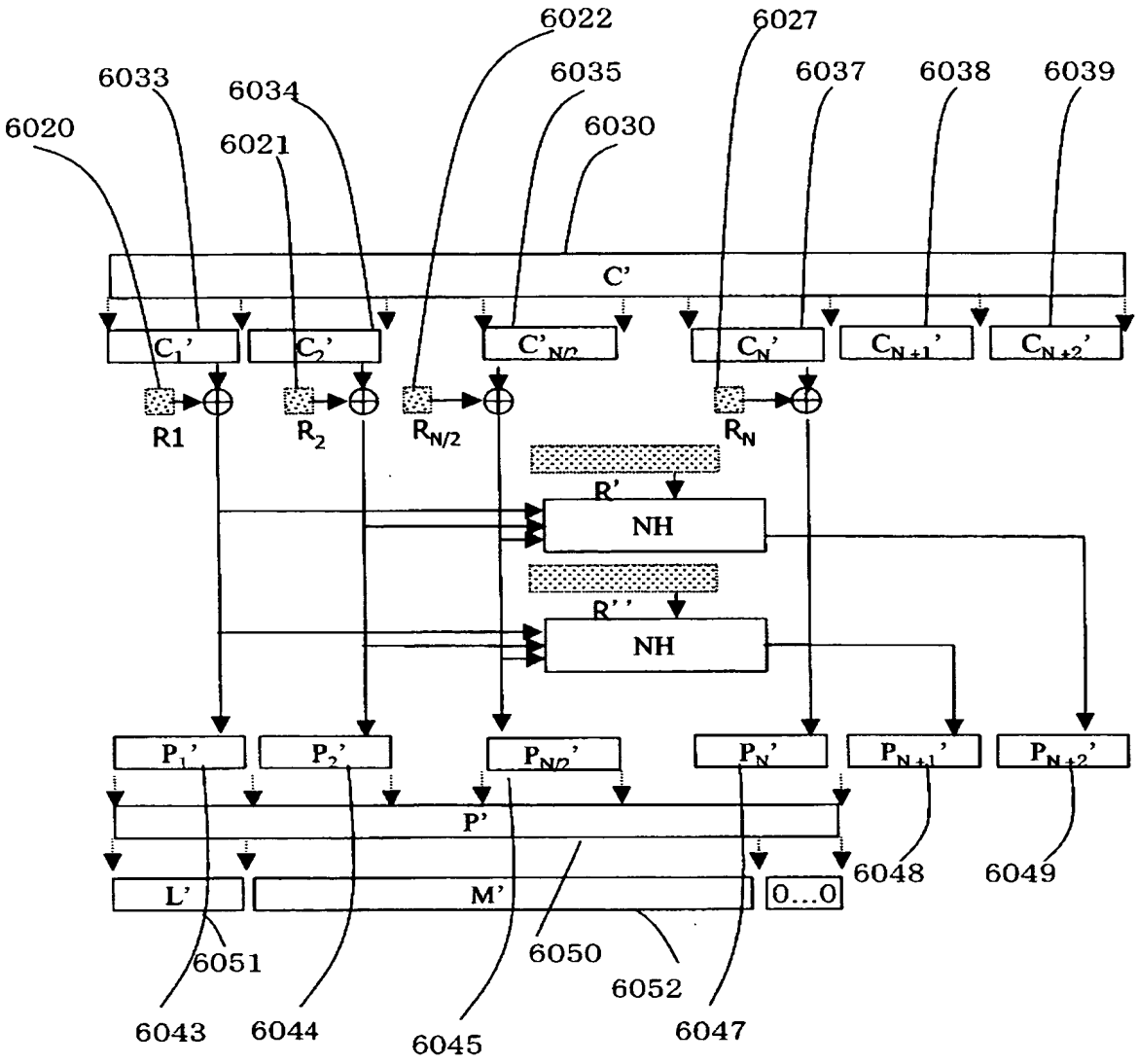
【図１５】

図15


5920
5921
5934
5928
5927
5935
5936
5937
5938
5930
5931
5932
L
M
0...
P
P1
P2
PN/2
PN
R1
R2
RN
R'
NH
R''
NH
C1
C2
CN/2
CN
CN+1
CN+2
C
5943
5944
5945
5956
5947
5948
5949
5929

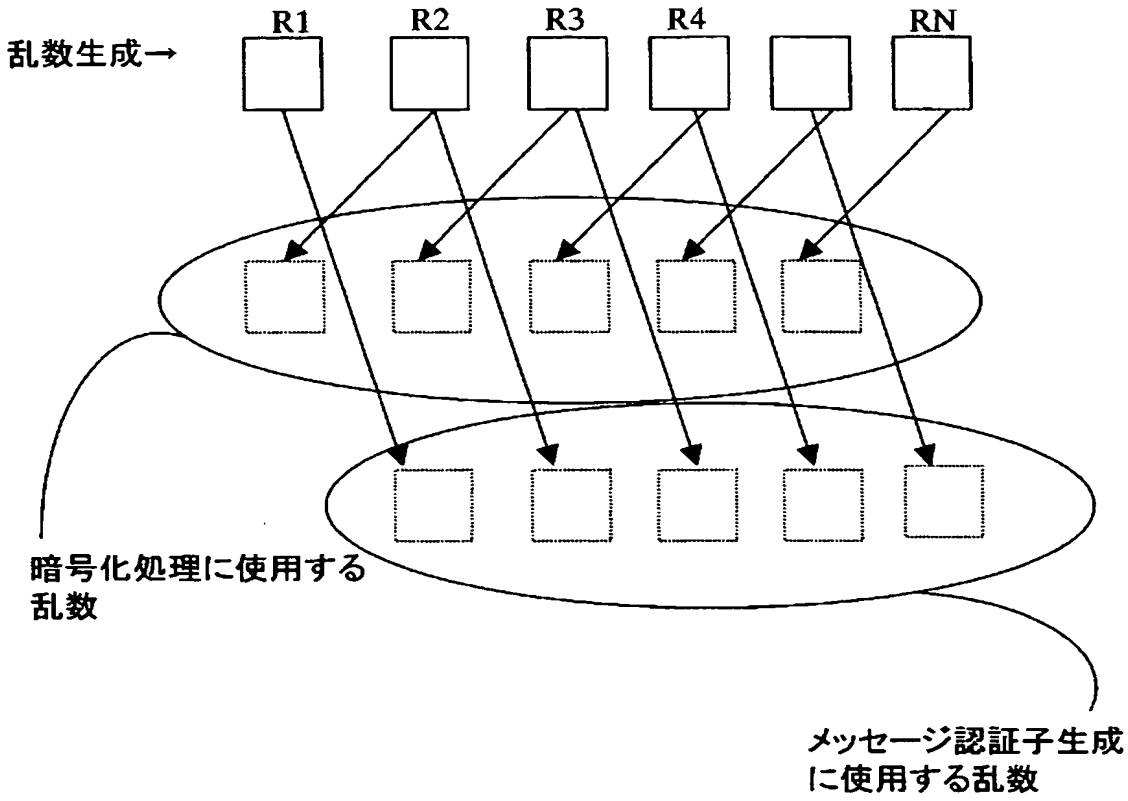
【図１６】

図16


6022
6027
6033
6034
6035
6037
6038
6039
6020
6021
6030
C'
C1'
C2'
C'N/2
CN'
CN+1'
CN+2'
R1
R2
RN/2
RN
R'
NH
R''
NH
P1'
P2'
PN/2'
PN'
PN+1'
PN+2'
P'
L'
M'
0...0
6048
6049
6051
6050
6052
6043
6044
6045
6047

【図１７】

図17



【書類名】要約書

【要約】

【課題】

従来の暗号技術では，復号の際，改ざん検出を行おうとする，異なるふたつの鍵共有の必要性，メッセージの２倍となる乱数の必要性，独立の処理，別の暗号学的要素関数の追加実装などが必要であった。

【解決手段】

乱数を生成して暗号化と認証処理を行い，事前計算と並列計算を達成する。また，生成する乱数の長さは，メッセージ長Nに対して2Nより少ない乱数を用いて，暗号処理と認証処理を行う。具体的には，疑似乱数生成器を使って乱数を生成しそれらをブロック毎に分割する。また平文もブロック毎に分割する。乱数ブロックと平文ブロックを排他的論理和し暗号文ブロックを得る。ハッシュ関数NHは，乱数ブロックを鍵入力とし，生成された暗号文のメッセージ認証子を生成する。乱数生成は事前計算可能であり，暗号文ブロック生成演算は並列処理可能であり，ハッシュ関数NHも並列処理可能であるため，高速計算ができる。

【選択図】図1

認定・付加情報

特許出願の番号	特願２００３－１５７４４４
受付番号	５０３００９２１７６１
書類名	特許願
担当官	第七担当上席　　００９６
作成日	平成１５年　６月　４日

＜認定情報・付加情報＞

【提出日】　平成15年 6月 3日

次頁無

特願２００３－１５７４４４

出　願　人　履　歴　情　報

識別番号　［０００００５１０８］

１．変更年月日　１９９０年　８月３１日
［変更理由］　新規登録
住　所　東京都千代田区神田駿河台４丁目６番地
氏　名　株式会社日立製作所